REMARKS/ARGUMENTS

This Amendment is in response to the Office Action dated August 29, 2006. A 2-month extension of time is submitted herewith. Claims 1-69 are pending. Claims 1-69 are rejected. Claims 1, 6, 17, 19, 22, 26, 27, 30, 32-37, 39-40, 44-51, 53-54, and 69 have been amended. Claims 1-15 and 55-68 have been canceled. Claim 70 has been added. Accordingly, claims 16-54 and 69-70 remain pending in the present application.

Attorney for the applicant acknowledges and appreciates the telephonic interview with Examiner Vu held on December 4, 2006. The amendments to the claims are a result of the comments made during the interview.

Claim Amendments

Independent claims 16, 36 and 69 have been amended, and claim 70 has been added, to more particularly claim the present invention. Claims 16 and 36 have been amended such that steps (a)-(d) recite how the computer program is prepared for obfuscation, while new step (e) recites what occurs when the computer program is executed on a computer. Independent claim 69 includes recitations similar to independent claim 16 and 36.

Step (a) has been amended to recite "partitioning the program into one or more non-critical code segments and one or more critical code segments" (as originally recited in claim 36). Step (b) has been amended to include "removing the critical code segments" and embedding the critical code segments within one or more exception handlers. Support for "removing" can be found in original claim 15 and in the Specification on page 12, lines 4+, for example. Reference to a debugger program has

been removed, as requested by the Examiner.

Steps (c) and (d) of claim 16 have been amended to make clear that a computer program is provided with an exception setup handler and in-line code segment for implementing execution of the exception handlers. Steps (c) and (d) of claim 36 similarly recite that the computer program is provided with a driver and an in-line code segment.

New step (e) recites "executing the computer program on a computer having an operating system that provides kernel-level and user-level execution modes, and debug resources to support generation and processing of exceptions at specified addresses" (as recited in original claim 69). Step (e) further recites "wherein execution flow of computer program is maintained with the non-critical code segments executing in the user-level protected mode, but when an address immediately prior to each one of the critical code segments in the execution flow is encountered, an exception occurs and the kernel-level execution mode is entered whereby control is transferred to the one or more exception handlers for execution of the critical code segments, and when the exception handlers complete execution, control is returned to the non-critical code segments." Support for this amendment can be found in the Specification on page 8, lines 13-17; and page 12, line 17 through page 13, lines 1-17, for example. Claim 70 includes similar recitations. Accordingly, no new is a matter has been entered.

§101 Rejection

The Examiner rejected claims 1-15 and 55-68 under 35 USC 101 because the claimed invention is directed to non-statutory subject matter. Although claims 1-15 and 55-68 have been canceled, Applicant respectfully traverses the rejection because

claims 1 and 55, as a whole, recite methods for increasing security of a software program by obfuscation of program execution flow by encapsulating (embedding) the critical code segments in respective exception handlers. Therefore, claims 1 and 55 positively recite a real-world application result that is concrete, tangible and useful, and thus are directed to statutory subject matter.

§103 Rejections

Claims 1-7, 15-18, 26, 36-38, 55-61, and 69 are rejected under 35 USC 103(a) as being unpatentable over Kuzara et al (5,450,586) and further in view of Held (5,889,988). Claims 8-14, 19-25, 27, 39-54, and 62-68 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuzara and Held, and further in view of Cardoza (5,630,049). Claims 28-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuzara and Held, further in view of Admitted Prior Art and Hagimont. Claims 32-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuzara, Held, and Hagimont, further in view of Cardoza (5,630,049). Applicant respectfully submits that the claims as amended are allowable over the references.

According to the exemplary embodiments, a method and system for increasing software program security by obfuscation of program execution flow is disclosed and claimed. The obfuscation of program execution flow hides key algorithms from view during debugging to increase the difficulty of reverse engineering, and to increase the difficulty of determining how to defeat anti-piracy features in the software. This is accomplished identifying critical code segments to be hidden in the software program, removing the critical code segments, and embedding them in exception handlers (Specification p. 12, lines 4 et seq.). When the program is executed, execution flow

proceeds as normal except that execution of the critical code segments is performed by exception handlers in kernel-mode of the operating system. A user-level debugger program is incapable of analyzing code executed within exception handlers kernel-level mode and will therefore be unable to provide information regarding the critical code segments of the software program for a hacker to reverse engineer. (Specification p. 10, lines 4-6)

In contrast, Kuzara discloses the debugging of an operating system "black box". By definition, one does not have insight or control over what happens in the black box. There is no ability to remove code segments of a software program into the black box. Because Kuzara does not have this ability, Kuzara discloses placing code markers at exit and entry points of the black box as interface service calls, but not within the black box itself.

The placement of code markers at the exit and entry points of a black box is not analogous to the removing of code intended to be executed at the user-level to code that is executed at the kernel-level, i.e., removing the critical code segments and embedding them in exception handlers. Kuzara does not disclose moving code from one execution level to another execution level.

Held fails to cure this deficiency of Kuzara. Therefore, Kuzara in view of Held does not teach or suggest removing the critical code segments and embedding the critical code segments in respective exception handlers, in combination the other elements as recited in amended claims 16, 36 and 69. Independent claims 16, 36 and 69 are therefore allowable over the references. Moreover, the remaining claims, which depend either directly or indirectly from one of claims 16 and 36 are considered allowable for at least these same reasons.

In view of the foregoing, it is submitted that claims 16-54 and 69-70 are allowable over the cited references. Because the secondary references stand or fall with the primary references, claims are allowable because they are dependent upon the allowable independent claims. Accordingly, Applicant respectfully requests re-consideration and passage to issue of claims 16-54 and 69-70 as now presented.

Applicants' attorney believes this application in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Respectfully submitted,
Strategic Patent Group

February 1, 2007

/Stephen G. Sullivan/
Stephen G. Sullivan
Attorney for Applicant(s)
Reg. No. 38,329
(650) 969-7474